1-28-03

Attorney Docket No.: 015685.P096 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:)
) Examiner: Not Yet Assigned
Orchard)
) Art Unit: 2122
Application No: 09/823,928)
)
Filed: March 31, 2001)
)
For: AN ARCHITECTURE AND RELATED)
METHODS FOR EFFICIENTLY)
PERFORMING COMPLEX ARITHMETIC)

Assistant Commissioner For Patents
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Sir:

Prior to examination of the above-referenced U.S. Patent Application, please enter this amendment.

IN THE SPECIFICATION

Please replace the paragraph beginning on page 9, line 9 with the following:

Fig. 3 illustrates a block diagram of an example extensible, hyperpipelined summing module architecture 304, in accordance with one example embodiment of the present invention. As introduced above, the innovative architecture of summing module 304 is dynamically implemented within one or more CLB (202) blocks of an FPGA by an instance of summing module generator 222. In accordance with the illustrated example implementation of Fig. 3, summing module 304 is depicted comprising a dynamically generated, pipelined hybrid Wallace adder tree 306 of one or more stages (extensible to a hyperpipelined Wallace tree, i.e., 306A-N) which feeds a final, m-input adder stage 318. For example, the final m-input adder stage 318 may be a stage of two-input adders adder stage 318. As shown, the hybrid Wallace tree 306A-N is presented comprising a dynamically determined number of full-adders (fa) and associated registers (R) 308, half-adders (ha) and associated registers (R) 310 and registers (R) 312. Those skilled in the art will appreciate that each of the hybrid elements are readily implemented within

one or more of a look-up table (LUT) and/or registers of a CLB slice of the FPGA, i.e., utilizing the atomic elements of an FPGA.

Please replace the paragraph beginning on page 10, line 4 with the following:

According to one implementation, summing module generator performs maximal segmentation (virtual grouping of bits denoted by dashed lines 303) within the column to group bits in groups of 3, 2, or 1 bit(s), respectively. Three-bit groups are passed to a full adder for processing, while two-bit groups are passed to a half-adder for processing. Single bit columns are passed directly to an available register 312 within a CLB. In accordance with one aspect of the invention, summing module generator 222 utilizes standard routing analysis tools to identify the optimal atomic layout of each of the allocated elements 308-312 of the hybrid Wallace tree 306. According to one implementation, summing module generator is designed to minimize waste of atomic resources and allocates elements 308-312 in this regard. According to one implementation, summing module generator 222 prioritizes performance speed over waste and, as a result, seeks to minimize routing among and between atomic elements 206-212 implementing the hybrid summing module 306, even at the expense of some waste of atomic resources. In another implementation, resource conservation and performance are equally weighted, with resources allocated accordingly.

Please replace the equation beginning on page 17, line 16 with the following:

(2) $I_2=(a_2*c_2)-(b_2*d_2)$ and added to I_1; performed simultaneously with $Q_2=(a_2*d_2)+(b_2*c_2)$ and added to Q_1.

IN THE CLAIMS

Please cancel claim 1.

Please add the following new claims:

2. (New) A method comprising:

analyzing input terms on a bit-wise basis;

selecting resources to generate the summing module based, at least in part, on the analysis; and

designing a hyperpipelined series of Boolean function generators to implement a Wallace-architecture of full-adders, half-adders, and associated registers in the selected